UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/824,998 | 04/14/2004 | Filipe Fortes | MS305925.1/40062.226US01 | 7282 |

27488 7590 03/07/2007

MERCHANT & GOULD (MICROSOFT)
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903

| EXAMINER | |
|---|---|
| TAN, ALVIN H | |
| ART UNIT | PAPER NUMBER |
| 2173 | |

| SHORTENED STATUTORY PERIOD OF RESPONSE | MAIL DATE | DELIVERY MODE |
|---|---|---|
| 3 MONTHS | 03/07/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/824,998 | FORTES ET AL. | |
| | Examiner | Art Unit | |
| | Alvin H. Tan | 2173 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *14 April 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 April 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *4/14/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

### *Remarks*

1. Claims 1-20 have been examined and rejected. This is the first Office action on the merits.

### *Claim Objections*

2. Claim 5 is objected to because of the following informalities:

a. On *[line 1]* of claim 5, Examiner suggests changing "comprises." to -- comprises:--

Appropriate correction is required.

### *Claim Rejections - 35 USC § 112*

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. Claims 1 and 5 recite the limitation "the parent" in *[line 5]* and *[line 6]* of claim 1 and *[line 3]* of claim 5. There is insufficient antecedent basis for

this limitation in the claim. Examiner suggests changing "the parent" to --the parent object--.

b. Claims 1, 4, and 5 recite the limitation "the child" in *[line 6]* of claim 1, *[line 3]* of claim 4, and *[lines 3 and 5]* of claim 5. There is insufficient antecedent basis for this limitation in the claim. Examiner suggests changing "the child" to --the child object--.

c. Claim 1 recites the limitation "the parameter" in *[line 7]* of the claim. It is unclear as to which of the one or more parameters associated with one of the child object and the parent object is being referred to.

d. Claim 1 recites the limitation "the one or more parameter limitations" in *[line 8]* of the claim. There is insufficient antecedent basis for this limitation in the claim.

e. Claim 3 recites the limitation "the object" in *[line 5]* of the claim. It is unclear whether "the object" refers to the child or parent object.

f. Claims 4 and 5 recite the limitation "the one or more child limitations" in *[line 2]* of the claims. There is insufficient antecedent basis for this limitation in the claim.

g. Claim 8 recites the limitation "the limitations" in *[line 9]* of the claim. It is unclear which of the one or more parameter limitations of the child, the one or more parameter limitations of the parent, or both, is being referred to.

h. Claim 8 recites the limitation "the received edit operation request" in *[line 10]* of the claim. There is insufficient antecedent basis for this limitation in the claim.

i. Claims 9, 10, 11, and 12 recite the limitation "the child" in *[line 3]* of claim 9 and *[line 2]* of claims 10, 11, and 12. There is insufficient antecedent basis for this limitation in the claim. Examiner suggests changing "the child" to --the child object--.

j. Claims 10, 11, and 12 recite the limitation "the parent" in *[line 2]* of the claims. There is insufficient antecedent basis for this limitation in the claim. Examiner suggests changing "the parent" to --the parent container--.

k. Claims 14 and 16 recite the limitation "the child" in *[line 6]* of claim 14 and *[lines 4 and 6]* of claim 16. There is insufficient antecedent basis for this limitation in the claim. Examiner suggests changing "the child" to --the child object--.

l. Claims 14 and 16 recite the limitation "the parent" in *[line 6]* of claim 14 and *[line 4]* of claim 16. There is insufficient antecedent basis for this limitation in the claim. Examiner suggests changing "the parent" to --the parent container--.

m. Claims 14 and 18 recite the limitation "the layout" in *[line 8]* of claim 14 and *[line 3]* of claim 18. There is insufficient antecedent basis for this limitation in the claim.

n. Claim 15 recites the limitation "the parent object" in *[line 3]* of the claim. There is insufficient antecedent basis for this limitation in the claim. Examiner suggests changing "the parent object" to --the parent container--

o. Claim 15 recites the limitation "the object" in *[line 6]* of the claim. It is unclear whether "the object" refers to the child object *[line 3]* of claim 14 or the parent object *[line 3]* of claim 15.

p. Claim 16 recites the limitation "the one or more child layout limitations" in *[line 3]* of the claim. There is insufficient antecedent basis for this limitation in the claim.

***Claim Rejections - 35 USC § 102***

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claim 1 and 14-16 are rejected under 35 U.S.C. 102(b) as being anticipated by Joseph (U.S. Patent No. 5,873,106).

**Claim 1 (Method)**

**Claim 14-16 (Machine-Readable Medium)**

6-1. Regarding claims 1 and 14, Joseph teaches the claim comprising detecting a layout edit operation for a child object displayed on a video display by a computer system, by disclosing a geometry management system that handles resizing of a window including the child objects and parent containers within the window *[column 1, lines 58-67; column 2, lines 1-11]*. The geometric system supports dynamic re-layout, wherein the size of one or more objects are changed at run time due to a variation in one or more parameters that affect the geometry management of the layout *[column 3, lines 19-24]*. As described in *[column 1, lines 38-55; figures 1a, 1b]*, an object may require resizing depending on the spacing of text and graphics. Thus, a layout edit operation for a child object is detected based on parameters that affect the geometry management of the layout. Additionally, the geometry management system's visual layout mechanisms permit a user to receive direct feedback when generating a display *[column 3, lines 35-42]*. At design or layout time, child objects are placed in a parent container *[column 1, lines 65-66]*. To lay out a form, a user selects a child object, drags the child object over a selected cell, and drops the child object into the cell. In response to the user operation, the geometry management system configures the child object within the parent container such that the appropriate geometry management methods are executed *[column 5, lines 40-48]*. As described in *[column 7, lines 34-48]*, parameters control the layout of child objects. Thus, during design time, a user may edit the layout of a child object by selecting its location as well as modifying its appearance by editing parameters of its associated geometry manager.

Joseph teaches determining from the child object and the parent whether there exists one or more parameters associated with one of the child and the parent, by disclosing that prior to design, a user specifies constraints that define relationships for the display layout. A constraint may be related directly to a child object, or may be a global constraint implemented at the geometry management level. A layout constraints attribute is set by a parent container object to control child objects within it *[column 4, lines 33-50]*. A child may also express preferences via minimum and maximum attributes for inner and outer sizes and via implementation of a QueryGeometry method *[column 5, lines 1-6]*. Both the parent and child parameters are taken into consideration when determining the geometry of the child object *[column 4, lines 51-56]*.

Joseph teaches editing the layout of the child object if a parameter limitation exists for the parameter, in accordance with the one or more parameter limitations, by disclosing that when a child object requires resizing, the child object requests geometry from the parent container object. Based on the specified parameters, the container determines whether a given geometry is feasible and if so, the container implements the geometry *[column 4, lines 56-67]*.

6-2. Regarding claim 15, Joseph teaches the claim wherein the determining further comprises determining a container type for the parent object or container in which the child object is displayed, by disclosing the layout constraints attribute, which stores container/geometry manager specific attributes within a child object *[column 9, Table 2, lines 24-31]*.

Joseph teaches retrieving a set of layout parameters related to the child object to be edited, by disclosing that a child may express preferences via minimum and maximum attributes for inner and outer sizes and via implementation of a QueryGeometry method *[column 5, lines 1-6]*. These parameters are used to determine the re-layout of the child object.

Joseph teaches retrieving a set of layout parameters related to the container in which the object is displayed, by disclosing that the layout constraints attribute is set by a parent container object to control child objects within it *[column 4, lines 33-50]*.

Joseph teaches recognizing any layout limitations that exist within the set of layout parameters, by disclosing dimension attributes *[column 9, lines 32-44]*.

6-3. Regarding claim 16, Joseph teaches the claim wherein the editing of the layout of the child object comprises determining whether the one or more child layout limitations includes a functional relationship between the child and parent, by disclosing an IsManaged attribute which determines whether the child is geometry managed by the parent of the child *[column 9, lines 26-28]*.

Joseph teaches retrieving a ReferenceSize if a functional relationship exists and calculating new layout parameters for the child based on the functional relationship and the ReferenceSize, by disclosing a Resized method that performs re-layout of child objects based on the current geometry of the container *[column 10, lines 41-56]*. A layout constraints attribute controls the layout of the child object with respect to its

container *[column 3, lines 33-50]*. Thus the layout of child objects are based on the parameters of the container.

***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 2-7 and 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joseph (U.S. Patent No. 5,873,106) and Software Engineering Institute ("Layering ABAS", March 15, 2003), herein after, SEI.

**Claims 2-7 (Method)**

8-1. Regarding claim 2, Joseph teaches the invention substantially as claimed. See section 6-1. Joseph does not expressly teach the claim wherein the detecting operation is performed via an abstraction layer. SEI teaches that abstraction layers are used to provide portability layers for software systems that run on different operating systems and hardware platforms and to provide a common abstraction for communications *[paragraph 1]*. Since the geometry management system of Joseph is used in any environment using a windows system *[column 1, lines 11-20]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to make the

geometry management system as an abstraction layer to arrange and edit the layout of objects. This would allow the geometry management system to be made portable and more easily implemented in any application having a windows system.

8-2. Regarding claim 3, Joseph and SEI further teach the claim wherein the determining operation comprises determining a container type for the parent object or container in which the child object is displayed, by disclosing the layout constraints attribute, which stores container/geometry manager specific attributes within a child object *[Joseph, column 9, Table 2, lines 24-31]*.

Joseph and SEI teach retrieving a set of properties related to the child object to be edited, by disclosing that a child may express preferences via minimum and maximum attributes for inner and outer sizes and via implementation of a QueryGeometry method *[Joseph, column 5, lines 1-6]*. These properties are used to determine the re-layout of the child object.

Joseph and SEI teach retrieving a set of properties related to the parent container in which the object is displayed, by disclosing that the layout constraints attribute is set by a parent container object to control child objects within it *[Joseph, column 4, lines 33-50]*.

Joseph and SEI teach recognizing any limitations that exist within the set of properties, by disclosing dimension attributes *[Joseph, column 9, lines 32-44]*.

8-3. Regarding claim 4, Joseph and SEI further teach the claim wherein the operation of editing comprises determining whether the one or more child limitations includes a maximum dimension, by disclosing minimum and maximum attributes for inner and outer sizes of child objects *[Joseph, column 5, lines 1-6]*.

Joseph and SEI teach limiting adjustment of the dimension of the child to less than or equal to the maximum if the maximum dimension is present, by disclosing that if an object is out of a set range, the object calls set attribute calls to bring the dimensions within the permissible range *[Joseph, column 9, lines 32-44]*.

8-4. Regarding claim 5, Joseph and SEI further teach the claim wherein the editing operation comprises determining whether the one or more child limitations includes a functional relationship between the child and parent, by disclosing an IsManaged attribute which determines whether the child is geometry managed by the parent of the child *[Joseph, column 9, lines 26-28]*.

Joseph and SEI teach retrieving a ReferenceSize if a functional relationship exists and calculating new layout parameters for the child based on the functional relationship, by disclosing a Resized method that performs re-layout of child objects based on the current geometry of the container *[Joseph, column 10, lines 41-56]*. A layout constraints attribute controls the layout of the child object with respect to its container *[Joseph, column 3, lines 33-50]*. Thus the layout of child objects are based on the parameters of the container.

8-5. Regarding claim 6, Joseph and SEI further teach the claim wherein editing the child object comprises modifying one or more properties of the child object, by disclosing that the geometric system supports dynamic re-layout, wherein the size of one or more objects are changed at run time due to a variation in one or more parameters that affect the geometry management of the layout *[Joseph, column 3, lines 19-24]*. As described in *[Joseph, column 1, lines 38-55; figures 1a, 1b]*, an object may require resizing depending on the spacing of text and graphics. Thus, the size of a child object may be edited.

8-6. Regarding claim 7, Joseph and SEI further teach wherein editing the child object comprises modifying one or more properties of the parent object or container, by disclosing that the geometric system supports dynamic re-layout, wherein the size of one or more objects are changed at run time due to a variation in one or more parameters that affect the geometry management of the layout *[Joseph, column 3, lines 19-24]*.

**Claims 17-20 (Machine-Readable Medium)**

8-7. Regarding claim 17, Joseph teaches the invention substantially as claimed. See section 6-1. Joseph does not expressly teach the claim wherein the detecting operation is performed via an abstraction layer. SEI teaches that abstraction layers are used to provide portability layers for software systems that run on different operating systems and hardware platforms and to provide a common abstraction for communications

*[paragraph 1]*. Since the geometry management system of Joseph is used in any environment using a windows system *[column 1, lines 11-20]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to make the geometry management system as an abstraction layer to arrange and edit the layout of objects. This would allow the geometry management system to be made portable and more easily implemented in any application having a windows system.

8-8. Regarding claim 18, Joseph and SEI teach the claim wherein editing the child object further comprises determining whether a layout limitation of the child is a proportional relationship to the parent and if so, maintaining the proportional relationship between the layout of the object and the parent container, by disclosing that the geometry management policy may be to configure a container large enough to encapsulate a child object text *[Joseph, column 2, lines 4-8]*. Thus, the geometry management policy may maintain a relationship between the sizes of the parent and child object such that if the size of the child object varies, the parent object will vary in a manner that is dependent of the child object.

8-9. Regarding claim 19, Joseph and SEI teach the claim wherein editing the object comprises modifying one or more properties of the child object in a child measure helper routine in the abstraction layer, by disclosing that the geometry management system specifies parameters through negotiation among the child objects, the containers, and the high level geometry manager *[Joseph, column 4, lines 51-67]*.

8-10. Regarding claim 20, Joseph and SEI teach the claim further comprising modifying one or more properties of the child object in a child arrangement routine in the abstraction layer consistent with one or more limitations in the parent container, by disclosing that the geometry management system specifies parameters through negotiation among the child objects, the containers, and the high level geometry manager *[Joseph, column 4, lines 51-67]*. When a child object requires resizing, the child object calls a RequestGeometry method contained in the corresponding container object. The container determines whether a given geometry is feasible and if the parameters are acceptable, the container implements the geometry *[Joseph, column 4, lines 56-67]*.

9. Claims 8-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joseph (U.S. Patent No. 5,873,106) and Bray ("Application Programming Interface", February 2, 2003).

**Claims 8-13 (System)**

9-1. Regarding claim 8, Joseph teaches the claim comprising a processor and a memory coupled with and readable by the processor, by disclosing *[figure 7]*.

Joseph teaches detecting a layout edit operation request for a child object displayed on the video display by the computer system and sending an edit operation request to initiate layout editing of the child object, by disclosing a geometry

management system that handles resizing of a window including the child objects and parent containers within the window *[column 1, lines 58-67; column 2, lines 1-11]*. The geometric system supports dynamic re-layout, wherein the size of one or more objects are changed at run time due to a variation in one or more parameters that affect the geometry management of the layout *[column 3, lines 19-24]*. As described in *[column 1, lines 38-55; figures 1a, 1b]*, an object may require resizing depending on the spacing of text and graphics. Thus, a layout edit operation for a child object is detected based on parameters that affect the geometry management of the layout. Additionally, the geometry management system's visual layout mechanisms permit a user to receive direct feedback when generating a display *[column 3, lines 35-42]*. At design or layout time, child objects are placed in a parent container *[column 1, lines 65-66]*. To layout a form, a user selects a child object, drags the child object over a selected cell, and drops the child object into the cell. In response to the user operation, the geometry management system configures the child object within the parent container such that the appropriate geometry management methods are executed *[column 5, lines 40-48]*. As described in *[column 7, lines 34-48]*, parameters control the layout of child objects. Thus, during design time, a user may edit the layout of a child object by selecting its location as well as modifying its appearance by editing parameters of its associated geometry manager.

Joseph teaches determining whether the child object has one or more parameter limitations and determining whether the parent container has one or more parameter limitations, by disclosing that prior to design, a user specifies constraints that define

relationships for the display layout. A constraint may be related directly to a child object, or may be a global constraint implemented at the geometry management level. A layout constraints attribute is set by a parent container object to control child objects within it *[column 4, lines 33-50]*. A child may also express preferences via minimum and maximum attributes for inner and outer sizes and via implementation of a QueryGeometry method *[column 5, lines 1-6]*. Both the parent and child parameters are taken into consideration when determining the geometry of the child object *[column 4, lines 51-56]*.

Joseph teaches editing the child object layout based on the limitations and the received edit operation request, by disclosing that when a child object requires resizing, the child object requests geometry from the parent container object. Based on the specified parameters, the container determines whether a given geometry is feasible and if so, the container implements the geometry *[column 4, lines 56-67]*.

Joseph does not expressly teach sending an edit operation request via an application program interface to initiate layout editing of the child object. Bray teaches that an Application Programming Interface (API) facilitates exchanging messages or data between two or more different software applications. This is used to achieve total cross-platform consistency. Since the geometry management system of Joseph is used in any environment using a windows system *[column 1, lines 11-20]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to make the geometry management system as an API to arrange and edit the layout of objects. This

would allow the geometry management system to be made portable and more easily implemented in any application having a windows system.

9-2. Regarding claim 9, Joseph and Bray teach the claim of performing a child object measure helper operation and a child object arrangement helper operation on the child when a layout edit operation request is detected, by disclosing that the geometric system supports dynamic re-layout, wherein the size of one or more objects are changed at run time due to a variation in one or more parameters that affect the geometry management of the layout *[Joseph, column 3, lines 19-24]*. As described in *[Joseph, column 1, lines 38-55; figures 1a, 1b]*, an object may require resizing depending on the spacing of text and graphics. The layout of the child object is changed based on constraints specified by a user prior to design that define relationships for the display layout. A constraint may be related directly to a child object, or may be a global constraint implemented at the geometry management level. A layout constraints attribute is set by a parent container object to control child objects within it *[Joseph, column 4, lines 33-50]*. A child may also express preferences via minimum and maximum attributes for inner and outer sizes and via implementation of a QueryGeometry method *[Joseph, column 5, lines 1-6]*. Both the parent and child parameters are taken into consideration when determining the geometry of the child object *[Joseph, column 4, lines 51-53]*.

9-3. Regarding claim 10, Joseph and Bray teach the claim wherein one or more of the child limitations includes a functional relationship of size between the child and the parent, by disclosing that when a child object requires resizing, the child object calls a RequestGeometry method contained in the corresponding container object. The container determines whether a given geometry is feasible and if the parameters are acceptable, the container implements the geometry *[Joseph, column 4, lines 56-67]*.

9-4. Regarding claim 11 and 12, Joseph and Bray teach the claim wherein the functional relationship is a proportional relationship between the child and the parent and wherein editing the layout of the child object comprises maintaining the proportional relationship between the child and the parent, by disclosing that the geometry management policy may be to configure a container large enough to encapsulate a child object text *[Joseph, column 2, lines 4-8]*. Thus, the geometry management policy may maintain a relationship between the sizes of the parent and child object such that if the size of the child object varies, the parent object will vary in a manner that is dependent of the child object.

9-5. Regarding claim 13, Joseph and Bray teach the claim wherein editing the child object comprises modifying one or more layout properties of the parent container, by disclosing that the geometry management policy may be to configure a container large enough to encapsulate a child object text *[Joseph, column 2, lines 4-8]*.

***Conclusion***

10. The prior art made of record on attached form PTO-892 and not relied upon is considered pertinent to applicant's disclosure. Applicant is required under 37 C.F.R § 111(c) to consider these references fully when responding to this action. The documents cited therein teach similar systems for layout editing operations of display objects in a graphical user interface.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Alvin H. Tan whose telephone number is 571-272-8595. The examiner can normally be reached on Mon-Thu 9:30-7 and alternating Fridays 9:30-6.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine Kincaid can be reached on 571-272-4063. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AHT
Assistant Examiner
Art Unit 2173

TADESSE HAILU
*Patent Examiner*